

Arquitectura para el Despliegue del servicio de Video bajo Demanda de IPTV, apoyada en Interactividad y Sistemas de Recomendaciones

Architecture for the Video on Demand service deployment on IPTV, based on Interactivity and Recommender Systems

Diego Fabián Durán Dorado

Ingeniero en Electrónica y Telecomunicaciones,
Universidad del Cauca
Estudiante Maestría en Ingeniería Telemática,
Universidad del Cauca
Investigador Grupo GIT, Universidad del Cauca
Popayán, Colombia
dduran@unicauca.edu.co

José Arciniegas

Profesor titular, Departamento de Telemática,
Universidad del Cauca
Investigador Grupo GIT, Universidad del Cauca
Popayán, Colombia
jlarci@unicauca.edu.co

Resumen— En este artículo se presenta una alternativa de solución arquitectónica para el despliegue y soporte del servicio de VoD (Video bajo Demanda) de contenidos de televisión en IPTV (Televisión sobre el Protocolo de Internet). Esta alternativa integra el soporte de los servicios interactivos wikis, chat, blogs, y sistemas de recomendaciones con enfoques basados en contenido, colaborativos e híbridos. La arquitectura se describe en diferentes vistas así: de despliegue, de componentes y dinámica a partir de un escenario específico. A diferencia de otras arquitecturas, la solución presentada facilita la convergencia en un entorno de televisión, de estándares y tecnologías abiertas de la Web 2.0 como los Servicios Web (Web Services), REST (Transferencia de Estado Representacional) y XML (Lenguaje de Marcas Extensible), lo cual es acorde con la tendencia actual en la Web de permitir la interoperabilidad entre aplicaciones software independientes. Un grupo de pruebas sobre un prototipo de la arquitectura permitió concluir que es posible obtener un buen rendimiento en cuanto a tiempos de respuesta en un entorno web real durante situaciones de estrés. En cuanto al ancho de banda, el servicio puede desplegar toda la funcionalidad sobre las velocidades de acceso ofrecidas en la actualidad, considerando que la arquitectura está diseñada para redes no gestionadas, por lo que no se asegura QoS.

Palabras clave— Arquitectura, Interactividad, IPTV Recomendaciones, VoD.

Abstract—This paper presents an alternative architectural solution for the deployment and support of the VoD (Video on Demand) service for television content on IPTV (Internet Protocol Television). This alternative integrates support of interactive services like wikis, chat, blogs and recommender systems with item based approaches, collaborative and hybrid. The architecture is described in different views like this: deployment, components and dynamic, based on a specific scenario. Unlike other architectures, this solution facilitates the convergence of standards and open technologies of Web 2.0 like Web

Services, REST (Representational State Transfer) and XML (Extensible Markup Language), which is consistent with the current trend on the Web to allow interoperability between independent software applications. From a set of tests concluded that it is possible to obtain a good performance in terms of response times in a real web during stress situations. As to bandwidth, the service can display all of the functionality on access speeds currently offered, considering that the architecture is designed to unmanaged networks therefore not assured QoS.

Keywords— Architecture, Interactivity, IPTV, Recommender, VoD.

I. INTRODUCCIÓN

A través del servicio de VoD (Video bajo Demanda) de IPTV (Televisión sobre Protocolo IP), los usuarios pueden elegir sus productos desde una lista o catálogo en la que pueden encontrar información relacionada, seleccionar el contenido de su interés, descargarlo y visualizarlo en un televisor analógico o digital, o un computador [1]. Con el tiempo, la cantidad de contenidos aumentó y los problemas propios de la tecnología y de usabilidad aparecieron causados por la saturación de información. Entre ellos: retardos por cambio de canal [2], baja probabilidad de encontrar contenidos de interés para los usuarios [3]-[5], categorización de contenidos insuficiente para gustos variados y específicos [3]-[8], y debido a sugerencias de usabilidad de televisión digital [9], la cantidad de información presentada en pantalla es reducida y los contenidos pueden perder visibilidad. Debido a la experiencia negativa de los usuarios

causadas por estos problemas [3], es necesaria la implementación de estrategias para faciliten el acceso a información personalizada que corresponda a las preferencias de cada persona [10].

De forma general, el intercambio de información especializada es la tendencia en IPTV [1]. Esto demostrado en entornos como el social [11] (*Social TV*), el comercial (*tCommerce*) [12], y el gobierno (*tGovernment*) [13], en los que el intercambio de conocimiento producido por la interacción entre usuarios se lleva a cabo a través de conjuntos de aplicaciones y servicios interactivos de apoyo.

Para buscar soluciones a los problemas anteriores, se presenta una propuesta arquitectónica que integra servicios interactivos y sistemas de recomendaciones como apoyo al servicio de VoD sobre una red no gestionada. La arquitectura tiene en cuenta la cadena de negocio actual de IPTV [14], las tecnologías y protocolos disponibles en la Web 2.0 y las técnicas de filtrado de información de los sistemas de recomendaciones [15].

El trabajo está basado en las experiencias obtenidas en el laboratorio de TDi (Televisión Digital Interactiva) de la Universidad del Cauca (Colombia) en el marco del proyecto ST-CAV [16], [17] (Servicios de *T-Learning* para el soporte de Comunidades Académicas Virtuales), no sólo en el campo de IPTV sino también en el de la Televisión Digital Terrestre y Móvil; y en el laboratorio de TDi de la Universidad de Vigo -España- donde se investigó y profundizó de forma parcial lo relacionado con los sistemas de recomendaciones.

II. CONTEXTO

A continuación se definen aspectos importantes en los que se enmarca este trabajo y los conceptos más relevantes relacionados con el marco teórico de esta investigación.

A. IPTV

El *Open IPTV Forum* contempla la prestación de servicios de IPTV tanto en redes gestionadas como en no gestionadas. En general, es un conjunto de servicios multimedia (televisión, video, audio, texto, gráficos y datos) distribuido por una red IP. Sobre una red gestionada debe poseer calidad de servicio, seguridad, interactividad y fiabilidad. Sobre una red no gestionada no se asegura calidad de servicio [18].

B. VoD

Servicio de IPTV que permite a los usuarios elegir contenidos de televisión (programas y películas, entre otros) desde una EPG (Guía Electrónica de Programación) [18].

C. Servicios Interactivos

En el marco del presente trabajo, se refiere a servicios que fomentan la interacción entre el usuario y el sistema de VoD. Pueden clasificarse en dos tipos así: 1) asociados y 2) no asociados al contenido. Los asociados extienden la información presentada en cada contenido o se refieren específicamente a uno en particular. Los no asociados no necesariamente extienden la información tratada en un contenido. Ejemplo del primer tipo es el servicio de “valoraciones” que permite al usuario calificar la calidad de un contenido. Esta calificación se llamará en adelante como DOI (Grado de Interés). En el segundo tipo se encuentran servicios que están de acuerdo a la tendencia de la Web 2.0 como: *chat*, noticas, foros y *wikis*, entre otros, y sirven para la circulación de información y generación colaborativa de conocimiento.

D. Sistemas de Recomendaciones

En el contexto de la televisión, son herramientas que generan sugerencias de contenidos de televisión a partir de las preferencias y opiniones de los usuarios. Son útiles durante el proceso de filtrado de la gran cantidad de contenidos disponibles y asistir a los usuarios en la búsqueda y recuperación de información. En cuanto a su forma de operar, se definen en la literatura tres enfoques que son los más utilizados actualmente en el entorno de la televisión digital: los basados en contenido, colaborativos e híbridos.

Los sistemas basados en contenido sugieren al usuario nuevos contenidos en función de su semejanza con otros que le han interesado en el pasado.

Los sistemas colaborativos predicen el DOI de un usuario por un contenido con base en los gustos y opiniones de otros usuarios con intereses similares.

Con el fin de obtener mejores recomendaciones, los sistemas híbridos explotan las ventajas de los otros enfoques y sobrellevan sus desventajas debido a su naturaleza complementaria [19].

E. TVAnytime

Especificación en la que participan algunas de las organizaciones más relevantes de la industria de creación y difusión de contenidos multimedia, que normaliza un formato de datos común para describir contenidos audiovisuales genéricos, instancias concretas de los programas, perfiles de usuario, información de segmentación de contenido y mecanismos para referenciar los programas con independencia de su localización y hora de emisión [20].

III. TRABAJOS RELACIONADOS

En [21] se propone una arquitectura para el servicio de VoD compuesta principalmente por un arreglo de nodos cuyo objetivo es atender la masiva demanda en IPTV. Para ello, se realizan copias de cada contenido en los nodos de una manera dinámica y se obtiene así una solución escalable para mejorar el uso del ancho de banda. Sin embargo, la arquitectura está enfocada en la distribución de flujos (*streaming*) por la red, y no se considera el despliegue de servicios interactivos, sistemas de recomendaciones u otras herramientas de apoyo importantes para mejorar la experiencia del usuario.

En [19] se desarrolla una metodología basada en la utilización de técnicas de inferencia de conocimiento de la web semántica para elaborar recomendaciones personalizadas y automáticas de contenidos de televisión. El objetivo de la metodología es descubrir relaciones semánticas entre las preferencias de los usuarios y los contenidos disponibles. Aunque esto implica ampliar la gama de recomendaciones, el sistema tiene un enfoque basado en contenido, por lo que pueden presentarse problemas de sobre-especialización. Este consiste en la recomendación sólo de contenidos similares entre sí y dejar de lado otros que podrían interesar. Es posible reducir sus efectos a través de la implementación de un enfoque híbrido.

En [22] se define una arquitectura y la implementación de un sistema de IPTV que permite compartir archivos multimedia. La arquitectura se centra en cubrir aspectos como la presentación de los archivos, el control de ejecución (adelantar, retroceder, pausar) y la adhesión de descripciones (de texto). La arquitectura se basa en el estándar IMS (Sub Sistema Multimedia IP) y se añade un

conjunto de funciones adicionales a las básicas de IMS. Este diseño dificulta su escalabilidad y convergencia con los *Web Services*, en caso de añadir servicios y aplicaciones de apoyo.

En [3] se describen los sistemas de recomendaciones de contenidos de televisión basados en filtrado colaborativo y en contenido para sistemas de VoD en IPTV. Este trabajo resalta la importancia de las recomendaciones en entornos con abundancia de información y pretende dar solución a los problemas asociados a la ineficiencia en la búsqueda de un contenido de interés para el usuario en catálogos grandes.

En los trabajos relacionados no se considera el uso de herramientas de apoyo para el acceso e intercambio de información entre usuarios como servicios interactivos y sistemas de recomendaciones en el servicio básico de VoD. Por otra parte, las arquitecturas descritas dificultan la convergencia de los estándares y tecnologías abiertas de la Web 2.0 como los Servicios Web (*Web Services*), REST (Transferencia de Estado Representacional) y XML (Lenguaje de Marcas Extensible), cuyo uso es acorde a la tendencia en la Web de permitir la interoperabilidad entre aplicaciones software independiente de la plataforma en la que están instalados.

IV. ARQUITECTURA ESTÁTICA

La arquitectura estática del sistema está conformada por dos vistas: la de “Despliegue” y la de “Nodos y componentes”. La primera, brinda una idea general del funcionamiento del servicio y la segunda explica en detalle cada uno de los nodos que conforman la arquitectura, sus componentes y funciones.

Entre los criterios de diseño de la arquitectura están:

- Es importante que sea posible la modificación de componentes sin afectar el funcionamiento del sistema y que a pesar de la gran carga computacional necesaria para cada proceso, el sistema se comporte eficazmente.
- La organización de nodos obedece a la cadena de negocio de IPTV actual.
- La arquitectura está orientada a los servicios y obedece a los principios de REST, por lo que cada servicio interactivo se concibe como un

recurso que puede ser consumido a través de peticiones GET y POST.

- Se considera un entorno IP no gestionado, por lo que los nodos de la arquitectura se interconectan a través de Internet.

A. Vista de despliegue

En la Fig. 1 se presenta la vista de despliegue de la arquitectura, en la que se delimitan claramente el “Proveedor de contenidos”, el “Proveedor de servicios” y el “Cliente”.

Cada uno de los nodos se describe en la Fig. 1.

1) En el Proveedor de contenidos

a. Información de los contenidos

Esta base de datos contiene la meta-información de los contenidos. Por ejemplo: Título, actores, sinopsis y palabras clave, entre otras.

b. Perfiles de usuario

Esta base de datos contiene la información de cada usuario.

c. Servidor de Contenidos

Se encarga de tener disponibles los contenidos que se transmiten por el canal IP.

d. Recomendador

Relaciona la información de los contenidos y los perfiles de usuario a través de algoritmos de filtrado para generar recomendaciones. Aunque el enfoque utilizado para el desarrollo del presente trabajo es basado en contenido, la arquitectura permite el uso de otros como colaborativo o híbrido.

2) En el Proveedor de Servicio de VoD

a. Servidor de Aplicación de IPTV

Contiene la aplicación de IPTV que se despliega en la pantalla del cliente y se encarga de coordinar interfaces, catálogos, contenidos, servicios interactivos y recomendaciones que componen el entorno de VoD. Además, monitorea la actividad de usuario para coleccionar información referente a sus gustos y utilizarla en la generación de recomendaciones.

b. Servidor de Servicios interactivos

Se encarga de recibir peticiones por parte del Servidor de Aplicación de IPTV generadas por el uso de los servicios interactivos por parte del usuario. Este nodo obedece a una arquitectura

orientada a servicios REST con intercambio de mensajes JSON (Notación de Objeto JavaScript).

3) En el Cliente

a. STB IPTV

Se instalan funciones *JavaScript* que se encargan de monitorear la actividad del usuario. Además, a través de su navegador permite el despliegue y navegación a través del servicio de VoD. En condiciones ideales, este dispositivo debe interpretar *JavaScript*, CSS (Hojas de estilo en Cascada) y HTML (Lenguaje de Marcado de Hipertexto).

B. Vista de Nodos y Componentes

En la Fig. 2 se muestra la vista en nodos y componentes de la arquitectura. Cada componente representa una unidad de software ejecutable.

Las tareas de cada componente se describen a continuación.

1) En el Servidor de Contenidos

a. Repositorio de Contenidos

Encargado de almacenar y poner a disposición para su consumo, los contenidos que hacen parte del conjunto del servicio de Video bajo demanda.

2) En el Servidor de Bases de Datos

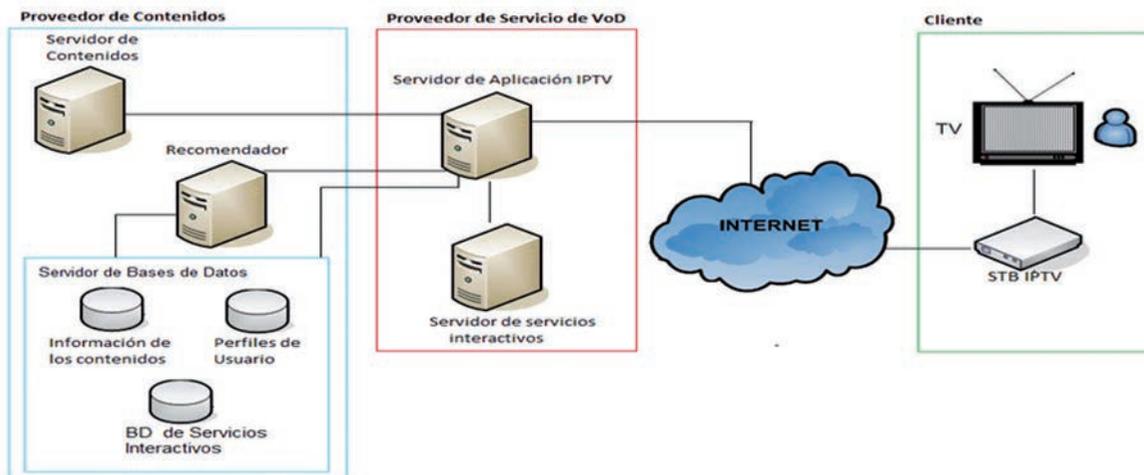
a. Información de los Contenidos

Esta base de datos contiene la meta-información de los contenidos. Por ejemplo: CRID (Identificador de Referencia de Contenido), título, actores, sinopsis, palabras clave, entre otras. Dependiendo del enfoque del sistema de recomendaciones, la información puede estar organizada en bases de datos relacionales o formalizada en una ontología que extrae las clases e instancias de la base de conocimiento OWL (Lenguajes de Ontologías Web). En cualquiera de los casos, la información responde a la especificación *TVAnytime*, que permite describir de manera ordenada cada contenido detalladamente y posibilita hallar de manera eficaz correlaciones entre ellos al momento de realizar recomendaciones.

b. Perfiles de usuario

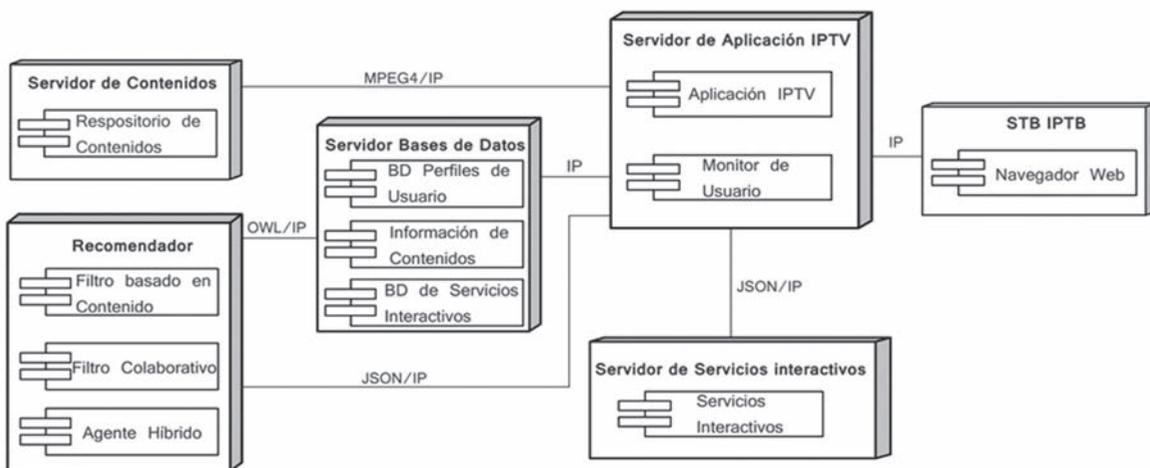
Esta base de datos contiene la información de cada usuario. En este contexto, contiene 4 tipos de información relevantes. El primero son las “Características demográficas” que abarcan datos como: sexo, edad, profesión, estado civil o número

Fig. 1. VISTA DE DESPLIEGUE DE LA ARQUITECTURA FUENTE: LOS AUTORES



Fuente: los autores

Fig. 2. VISTA EN NODOS Y COMPONENTES DE LA ARQUITECTURA



Fuente: los autores

ro de hijos; e información contextual como la localización del usuario, el horario en que prefiere ver televisión y el historial de visionado. El segundo, son “Matrices de valoraciones” que contiene el historial de DOI que el usuario ha asignado a cada contenido. El tercer tipo de información es una “Matriz de interactividad” que contiene información implícita registrada por el sistema mientras el usuario utiliza los servicios interactivos. Esta depende de los servicios ofrecidos y puede ser: tiempo de uso, con qué otros usuarios lo utiliza y qué tipo de contenido ve mientras lo usa, entre

otros. Finalmente, el cuarto tipo de información es una estructura perfil-ontología, en la cual el perfil del usuario está referido a todas las clases de la ontología (en caso de usarla) para conocer así sus preferencias.

c. BD de Servicios Interactivos

Contiene la información que manejan los servicios interactivos. Si el servicio es el *chat*, por ejemplo, contiene todos los mensajes, qué usuario los introdujo, la hora y fecha, entre otros. O si es valoraciones, registra los DOI, la fecha y el usuario que los generó.

3) En el Servidor de Servicios Interactivos

a. Servicios Interactivos

Encargado del soporte de los servicios interactivos. De acuerdo a REST y la actividad del usuario, pone los servicios interactivos a disposición para que sean consumidos como un recurso a través de peticiones GET y POST. De acuerdo a esto, registra o consulta información en la **BD de Servicios Interactivos** y envía respuestas a la **Aplicación de IPTV** en forma de mensajes JSON. Cada servicio interactivo que pertenece a esta componente, tiene su propio URI (*Uniform Resource Identifier*).

4) En el Recomendador

a. Filtro Basado en Contenido

Utiliza un algoritmo de filtrado basado en contenido sobre la información de los perfiles de usuario para recomendar a cada persona contenidos similares a los que le interesaron en el pasado.

b. Filtro Colaborativo

Utiliza un algoritmo de filtrado colaborativo basado en vecinos cercanos sobre la información del perfil de cada usuario para realizar recomendaciones de contenidos que interesaron a usuarios similares.

c. Agente Híbrido

Encargado de aplicar criterios de decisión para definir qué contenidos harán parte de la lista de recomendaciones. Esta puede estar compuesta por contenidos generados por el filtrado basado en contenido, el colaborativo o un porcentaje de cada uno.

5) En el Servidor de Aplicación de IPTV

a. Monitor de Usuario

Registra la actividad del usuario y almacena información relevante en las **BD Perfiles de Usuario** y **BD de Servicios Interactivos**

b. Aplicación IPTV

Organiza la información en interfaces que serán presentadas al usuario en pantalla. Según la actividad, obtiene información de contenidos y usuarios, además, envía peticiones GET y POST a los **Servicios interactivos** y desglosa los mensajes JSON recibidos para incorporarlos en las interfaces. Por otro lado, según la demanda de cada usuario, envía peticiones de contenidos al **Servidor de Contenidos** y los despliega en pantalla.

6) En el STB IPTV

a. Navegador Web

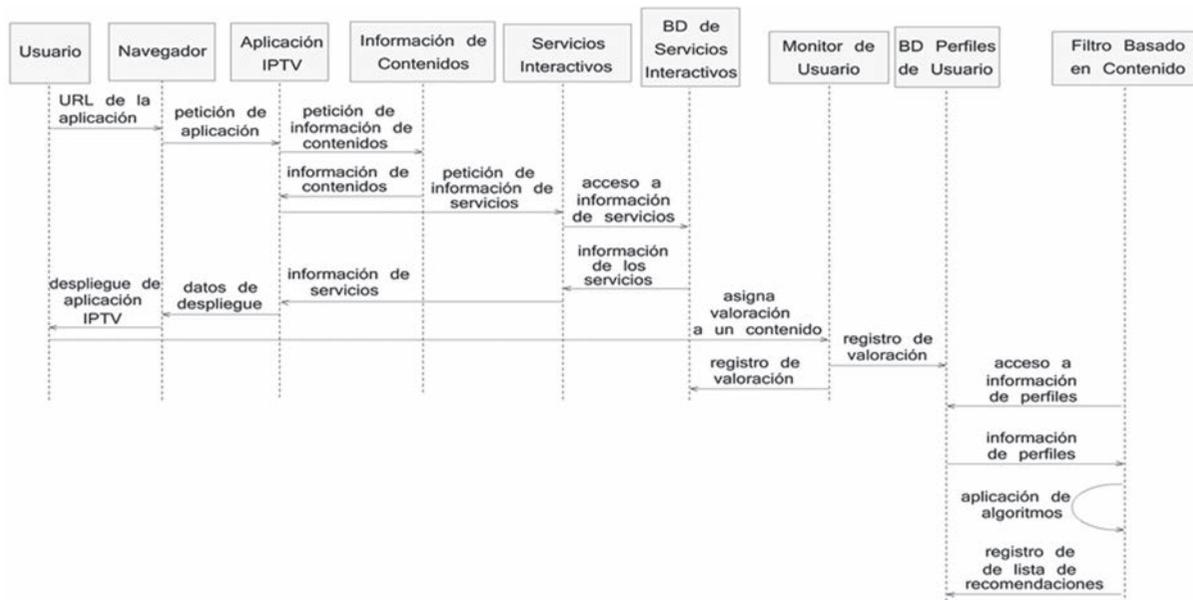
Navegador Web con el cual vienen equipados los STB y permite el despliegue y navegación por la aplicación.

V. ARQUITECTURA DINÁMICA

Una manera de ver la relación dinámica entre los componentes de la arquitectura es mediante un diagrama de secuencia. Para ello se ha seleccionado el siguiente escenario: un usuario accede al servicio de VoD, luego califica la calidad de un contenido a través del servicio de valoraciones, finalmente el sistema realiza el proceso para generar recomendaciones (ver Fig. 3).

El flujo de información inicia cuando el **Usuario** introduce la dirección URL de la **Aplicación IPTV**. La **Aplicación IPTV** realiza la petición de la información de los contenidos a la base de datos **Información de Contenidos** a través de sentencias SQL. Ahora la aplicación se despliega a través del navegador. El **Usuario** desea utilizar el servicio de valoraciones, por lo que la **Aplicación IPTV** realiza su invocación a través de peticiones GET, su URL y puerto a **Servicios Interactivos**. Los **Servicios Interactivos** realizan la petición de la información relacionada con el servicio de valoraciones a la base de datos **BD de Servicios Interactivos** a través de sentencias SQL. Esta información es recibida por la **Aplicación IPTV** en forma de mensajes JSON, que son desglosados y mostrados en pantalla a través de las interfaces gráficas. Con la navegación e interacción del usuario en el servicio, la información (de contenidos y servicios) presentada en pantalla se actualiza dinámicamente haciendo uso de *JavaScript* y *Ajax* (*JavaScript* y *XML Asíncronos*). Por ejemplo, si hay una nueva valoración de otro usuario, la nueva valoración promedio debe actualizarse. Cuando el Usuario realiza la valoración, el **Monitor de Usuario** registra el valor en la **BD de Servicios Interactivos** para que sea tenido en cuenta en los próximos usos y en la **BD perfiles de Usuario** para coleccionar información referente a sus gustos y realizar recomendaciones posteriormente. Para generar recomendaciones, el **Filtro Basado en Contenido** ejecuta algoritmos de filtrado sobre la información registrada en **BD Perfiles de Usuario** para generar una lista de recomendaciones, que posteriormente se registra en la **BD Perfiles de Usuario**.

Fig. 3. DIAGRAMA DE SECUENCIAS QUE RELACIONA LOS COMPONENTES A PARTIR DE UN ESCENARIO ESPECÍFICO



Fuente: los autores def GET(self, args): class recomendaciones: [http://servidor:puerto/recomendaciones/?idusuario=1 idusuario = web.input\('idusuario'\).idusuario](http://servidor:puerto/recomendaciones/?idusuario=1 idusuario = web.input('idusuario').idusuario)

VI. PROTOTIPO

A continuación se describe un prototipo de arquitectura que menciona lenguajes, plataformas, metodologías, tecnologías y protocolos utilizados y demuestra la interoperabilidad y convergencia que permite la arquitectura.

A. Servicios Interactivos

Este componente fue desarrollado en lenguaje Python en el entorno Eclipse en Linux. Para una buena configuración del entorno de desarrollo, es necesaria la instalación de: complemento *Pydev* para Eclipse, librerías *Python* de desarrollo, librerías de *networking Python web.py* y *gevent*, además del paquete de índice *Python Greenlet*. La aplicación es creada como un proyecto *PyDev* y se elige *Python* como intérprete. Según los principios de REST, en este componente se define cada servicio interactivo al especificar para él una URI. En el siguiente fragmento de código se define el servicio de recomendaciones, para el cual se ha especificado su URI. El servicio es consumido por la Aplicación IPTV mediante una petición GET que incluye datos como el Id del usuario para el cual se requiere consultar la lista de recomendaciones.

Con los datos recuperados a partir del URI puede definirse la lógica del servicio, por ejemplo para realizar consultas en bases de datos.

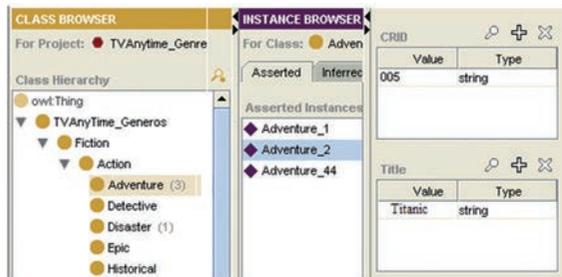
B. El Sistema de Recomendaciones

Es un sistema de recomendaciones basado en contenido, resultado de la adaptación de la metodología propuesta en [19]. Las características de dicho trabajo utilizadas en el presente estudio son: se basa en los DOI dados por cada usuario a cada contenido a través del servicio de valoraciones; los contenidos están organizados y formalizados en una ontología OWL, basada en la clasificación por “Géneros” de la norma *TVAnytime* en la que las clases hoja son los géneros; cada contenido está definido como individuo de varias clases hoja con dos propiedades de tipo *Datatype* llamadas CRID y Título. Como resultado del presente estudio, algunas características relacionadas con los contenidos – además relevantes para generar recomendaciones – pueden definirse dentro de la ontología con una clasificación diferente a la de los géneros. En este caso, la característica “Director” está formalizada en la ontología OWL como una clase hoja y sus individuos son los contenidos en los que ha sido director. Igual que como ocurre

con los géneros, cada individuo de cada director tiene las propiedades CRID y Título. En la Fig. 4, se presenta un fragmento de la clasificación por géneros de ontología desarrollada en la herramienta *Protégé* [23]. Aquí, la película *Titanic* tiene CRID igual a 005 y es individuo de la clase hoja Aventura (*Adventure*). En pro de las recomendaciones, también pertenece a las clases hoja Desastre (*Disaster*) y Drama Clásico (*Classical drama*). Así mismo, en la Fig. 5 se observa un fragmento de la clasificación por director, en la que la película *Titanic* es un individuo de la clase hoja James Cameron.

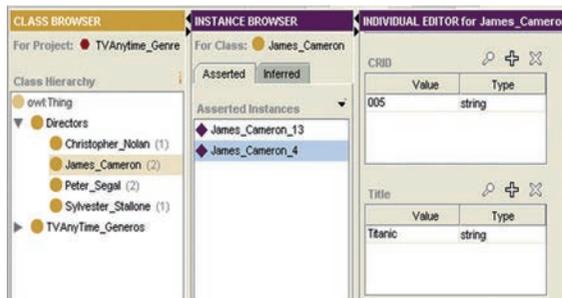
Para generar las recomendaciones, primero se realiza el cálculo del nivel de interés de las clases hoja del género y director para cada Usuario.

Fig. 4. FRAGMENTO DE LA CLASIFICACIÓN POR GÉNERO DE LA ONTOLOGÍA DESARROLLADA EN *PROTÉGÉ*



Fuente: los autores

Fig. 5. FRAGMENTO DE LA CLASIFICACIÓN POR DIRECTOR DE LA ONTOLOGÍA DESARROLLADA EN *PROTÉGÉ*



Fuente: los autores

Para esto, se promedian los DOI de todos los contenidos que pertenecen a una clase hoja, por medio de la expresión:

$$DOI_{px}(C_p) = \frac{1}{N} \cdot \sum_{i=1}^N DOI_{px}(P_i) \quad (i)$$

donde $DOI_{px}(C_p)$ es el nivel de interés del usuario con perfil px por la clase hoja C_p , N es el total de contenidos que pertenecen a la clase hoja C_p y $DOI_{px}(P_i)$ es el DOI del usuario con perfil px por el contenido P_i .

La lista de recomendaciones está compuesta por los contenidos pertenecientes a las clases hoja con mayor DOI para un usuario.

Se propone enriquecer la ontología con más características de clasificación como: "Actor principal", "País", "Año", "Actores de reparto", entre otros. De esta manera, se generan sugerencias basadas en los gustos de cada usuario y variadas en cuanto a temática, se soluciona así el problema de sobre-especialización de los sistemas de recomendaciones basados en contenido.

C. Información de los contenidos

La información de los contenidos es de dos tipos: meta-información y ontología OWL. La meta-información (CRID, Título, Sinopsis y actores, entre otros) es la que se presenta en pantalla cuando el usuario navega por los catálogos y está registrada en una base de datos relacional. La ontología OWL se utiliza para clasificar los contenidos según su género y otras características, como se indica en la sección VI-A. La ontología se especifica en un documento de texto plano con extensión OWL generado por *Protégé*. Este documento es consultado por otros componentes de la arquitectura para generar recomendaciones. Un fragmento de este documento se muestra a continuación, donde se especifica el género (*Adventure*) y CRID (005) de la película *Titanic*.

```
<Adventure rdf:ID="Adventure_2">
<Title
rdf:datatype="http://www.w3.org/2001/
XMLSchema#string"
>Titanic</Title>
<CRID
rdf:datatype="http://www.w3.org/2001/
XMLSchema#string"
>005</CRID>
</Adventure>
```

D. Perfiles de Usuario

Esta información está organizada en bases de datos relacionales. Además de almacenar información demográfica (Nombre, edad y género, entre otros) contiene información en una

estructura denominada perfil-ontología que modela los gustos del usuario relacionándolo con cada contenido y cada clase hoja de la ontología. En la Fig. 6 se presenta un fragmento de información perfil-ontología, donde está registrado el DOI asignado por diferentes usuarios a diferentes contenidos. Se puede observar que el usuario con Id 1 ha valorado el contenido con CRID 005 (*Titanic*) con un DOI igual a 5 y el 003 (Locos de ira) con un DOI igual a 3.

Fig. 6. FRAGMENTO DE INFORMACIÓN PERFIL-ONTOLOGÍA QUE RELACIONA CADA PERFIL Y SUS DOI POR CADA CONTENIDO

idUsuario	CridContenido	doi
1	001	1
1	005	5
2	003	3

Fuente: los autores

Los datos presentados en la Fig. 6 son utilizados para calcular el DOI de cada usuario por cada una de las clases hoja de la ontología a través de (1). En la Fig. 7 se presenta un fragmento de información perfil-ontología donde está registrado el DOI de cada usuario por cada una de las clases hoja de la clasificación por género. Aquí, el usuario con id 1 presenta un DOI por la clase hoja Desastre (*Disaster*) igual a 5 y por la clase hoja Aventura (*Adventure*) un DOI igual a 3. Se infiere que este usuario presenta un mayor interés por el género Desastre que por Aventura. Además, no ha valorado contenidos clasificados en el género Detective, por lo que no se puede inferir conclusión alguna.

Fig. 7. FRAGMENTO DE INFORMACIÓN PERFIL-ONTOLOGÍA DONDE SE RELACIONA CADA PERFIL Y SUS DOI POR CADA CLASE HOJA DE LA ONTOLOGÍA EN LA CLASIFICIÓN POR GÉNERO

idUsuario	Adventure	Detective	Disaster	Sports	Classical_drama
1	3	0	5	1	5
2	3	0	0	0	0

LOGÍA EN LA CLASIFICIÓN POR GÉNERO

Fuente: los autores

En la Fig. 8 se presenta otro fragmento perfil-ontología donde está registrado el DOI de cada usuario por las clases hoja de la clasificación por director.

Fig. 8. FRAGMENTO DE INFORMACIÓN PERFIL-ONTOLOGÍA DONDE SE RELACIONA CADA PERFIL Y SUS DOI POR CADA CLASE HOJA DE LA ONTOLOGÍA EN LA CLASIFICIÓN POR DIRECTOR

idUsuario	James_Cameron	Peter_Segal	Sylvester_Stallone
1	5	0	1
2	0	0	0
NULL	NULL	NULL	NULL

Fuente: los autores

E. Filtro Basado en Contenido

Este componente está desarrollado en lenguaje *Java* debido a la disponibilidad de librerías disponibles para la consulta de ontologías. Consulta en la **BD Perfiles de Usuario** los datos de la Fig. 6 y consulta la ontología haciendo uso del *framework* Jena [25] para descubrir a qué clases hoja pertenece cada contenido valorado. Después aplica (1) para obtener el DOI del usuario por las clases hoja de la ontología. Así se obtiene la información presentada en la Fig. 7 y Fig. 8. A partir de estos datos, se infiere el interés de cada usuario por los géneros, los directores y otras características que puedan ser formalizadas en la ontología y generar la lista de recomendaciones final. Por ejemplo, según la información presentada en la Fig. 7 y Fig. 8, se infiere que los contenidos recomendados deben ser en mayor proporción del género Desastre, menor de Aventura, y ninguna de Comedia. Así mismo, se pueden recomendar contenidos dirigidos por James Cameron y ninguno por Sylvester Stallone.

F. Aplicación IPTV

Este componente se desarrolló en el entorno de desarrollo *Netbeans* bajo la tecnología *Java JSP*. Las interfaces se definieron en *HTML*, añadiéndole dinámica a través de *JavaScript*, específicamente con la librería *Jquery*. Para mostrar información dinámica, es necesario realizar el consumo de los servicios interactivos a través de peticiones *GET* y *POST*. Debido a que *JavaScript* está del lado del cliente, y las peticiones a los servicios interactivos están del lado del servidor, es necesario el uso de técnicas *AJAX* para invocar *Servlets* que realicen las peticiones directamente. Por ejemplo, para la validación a través de usuario y contraseña primero se hace la invocación de un *Servlet* así:

```
$.ajax({
  url : "/vod/ValidaServlet",
  data: "login="+login+"&password="+password,
  type : "GET",
  cache : true,
  success : function(response) {} })
```

El *servlet* realiza la petición GET al componente de Servicios Interactivos, especificando el URI del servicio encargado de la validación de usuarios:

```
response=rc.get("http://"+servidor+":"+puerto+"/
validacion/?login="+login+"&password="+password);
```

Las respuestas enviadas por el componente de Servicios interactivos tienen formato JSON, por lo que es necesario desglosarlas para recuperar la información útil que será presentada en pantalla. Para esto se hace uso de *Jquery*:

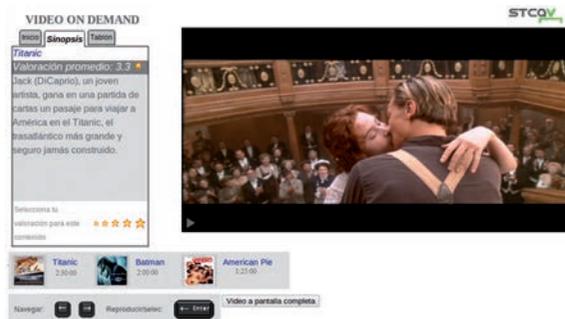
```
Var respuesta
= jQuery.parseJSON(response);
exito =respuesta.exito;
nombre = respuesta.Nombre;
idusuario=respuesta.id;
```

VII. APLICACIÓN FINAL DE USUARIO

A continuación se describe de forma general el uso de la aplicación de usuario. Una vez el usuario ingresa el URI de la aplicación a través del navegador Web, debe ingresar los datos de autenticación: usuario y contraseña. Si los datos de autenticación son correctos, se notifica que el usuario se ha autenticado correctamente.

La Fig. 9 muestra la interfaz del prototipo que presenta la organización de servicios, catálogos y reproducción de contenidos.

Fig. 9. INTERFAZ DEL PROTOTIPO QUE PRESENTA LA ORGANIZACIÓN DE SERVICIOS, CATÁLOGOS Y REPRODUCCIÓN DE CONTENIDO



Fuente: los autores

La Fig. 10 muestra la interfaz que presenta el servicio interactivo del chat.

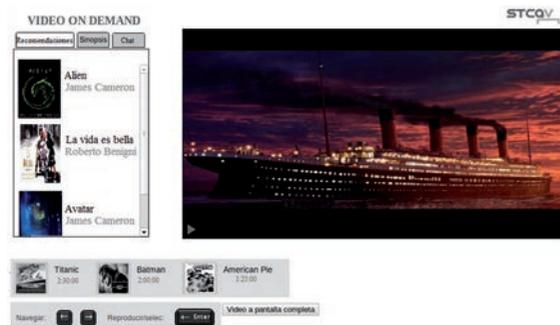
La Fig. 11 muestra la interfaz del servicio de recomendaciones.

Fig. 10. INTERFAZ QUE PRESENTA EL SERVICIO INTERACTIVO DE CHAT



Fuente: los autores

Fig. 11. INTERFAZ QUE PRESENTA EL SERVICIO DE RECOMENDACIONES



Fuente: los autores

VIII. PRUEBAS SOBRE EL PROTOTIPO

A. Entorno de pruebas

Para la realización de pruebas, se configuró una red como la mostrada en la Fig. 12. Un equipo es el servidor de la Aplicación IPTV, otro es el encargado de generar tráfico IP en la red con la herramienta *Ostinato*, otro es el monitor encargado de capturar datos con *Wireshark*, y el último es el cliente que realiza el consumo de la aplicación y obtiene datos de tiempo de respuesta con *Jmeter*.

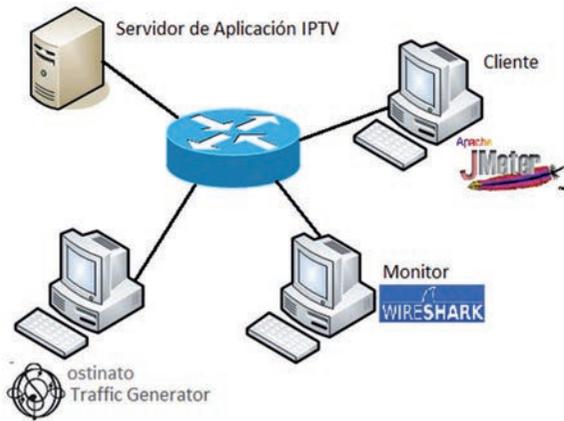
Con esta configuración, se realizó un conjunto de pruebas de rendimiento y de ancho de banda sobre el prototipo de la arquitectura propuesta.

B. Pruebas de Rendimiento

Se realizó dos pruebas de estrés relacionadas con el rendimiento del sistema con el fin de pre-

decir su tiempo de respuesta frente a un conjunto de peticiones *GET* de acuerdo a los principios de *REST*. Al hacer uso de la herramienta *Jmeter*, se configuraron las pruebas de la siguiente manera: 1) se simuló un grupo de 50 usuarios que al tiempo realizaron 3 peticiones *GET* cada uno; 2) se simuló un grupo de 100 usuarios que al tiempo realizaron 3 peticiones *GET* cada uno.

Fig. 12. CONFIGURACIÓN DE LA RED PARA PRUEBAS



Fuente: los autores

Para las dos pruebas se configuró *Jmeter* de tal forma que las tres peticiones consistieron en validación, petición de los mensajes del *chat* y petición de la lista de recomendaciones. El análisis de los resultados obtenidos con *Jmeter* se hace calculando un intervalo de confianza con un nivel de 95%, suponiendo una distribución Normal de la población. Los datos de la primera prueba se presentan en la Fig. 13.

Fig. 13. RESULTADOS DE RENDIMIENTO CON JMETER

Label	# Muestras	Media	Mediana
HTTP Request H...	24000	552	499
TOTAL	24000	552	499

Std. Dev.	% Error	Rendimiento
110,23	0,00%	178,1/sec
110,23	0,00%	178,1/sec

Fuente: los autores

En la primera prueba, el servidor respondió a los 50 hilos con un rendimiento de 181,7 peticiones por segundo. Así mismo, el tiempo promedio para recibir una respuesta del servidor es de 270 milisegundos, realizándose un total de 12000 peticiones *GET*. Con los datos de la Fig. 13, se calcula un intervalo de confianza determinado por (2),

el cual define los límites entre los que se espera una respuesta del servidor para una cantidad de usuarios que realiza peticiones de forma simultánea.

$$[TP - Z_{0,95} * D / \sqrt{n}, TP + Z_{0,95} * D / \sqrt{n}], \quad (2)$$

donde: tiempo promedio de respuesta $TP=270$ ms; $Z_{0,95}=1,96$; desviación estándar $D=64,57$; número de muestras $n=12.000$. Al remplazar los valores en (2) se obtuvo el intervalo de confianza:

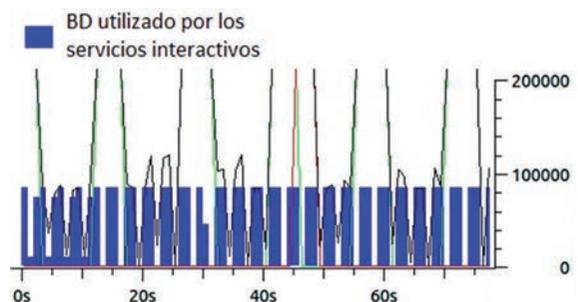
[268,8447; 271,1553] en milisegundos. En la segunda prueba y con un procedimiento como el anterior, se obtiene un intervalo de confianza: [550,6053; 553,3946] en milisegundos. Según los resultados, el uso de los principios de *REST* en la arquitectura permite la implementación de servicios interactivos que bajo situaciones de estrés presentan un buen rendimiento, si se tiene en cuenta los límites relacionados con el tiempo de respuesta de una aplicación web [26]: 1 segundo es el límite en el cual el usuario siente que navega libremente sin esperar demasiado una respuesta del servidor.

C. Pruebas de Ancho de Banda

Sobre el terminal de monitoreo se hizo un análisis de ancho de banda, con los datos de paquetes capturados por la herramienta *Wireshark*.

La Fig. 14 corresponde al ancho de banda generado por la validación, petición de los mensajes de *chat*, y petición de la lista de recomendaciones. Se observa que se realiza un consumo periódico de ancho de banda de aproximadamente 80 Kbps que equivale al 2% de la velocidad del enlace (5 Mbps). La periodicidad es generada por la comunicación continua entre el cliente y el servidor, con el fin de establecer una actualización continua de información en servicios como el *chat* o las valoraciones promedio.

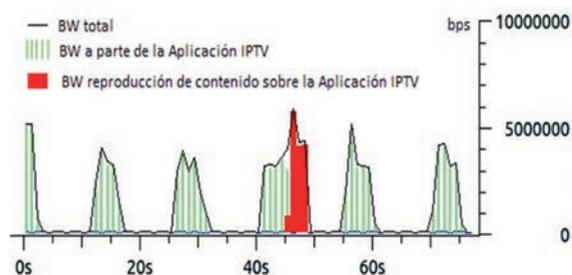
Fig. 14. ANCHO DE BANDA UTILIZADO POR EL USO DE LOS SERVICIOS INTERACTIVOS



Fuente: los autores

La Fig. 15 muestra la relación entre el ancho de banda total de la red, el ancho de banda utilizado por otros servicios aparte de la Aplicación IPTV y el utilizado cuando se reproduce un contenido en la aplicación. Se observa que la reproducción del contenido utiliza casi la totalidad del ancho de banda disponible, mientras se reduce el utilizado por otros servicios. Este resultado no se obtendrá siempre que se reproduzca un contenido en la Aplicación IPTV, debido a que está diseñada para una red no gestionada que no asegura QoS. Es decir, en algunas situaciones el ancho de banda de otros servicios hará que se reduzca el de la Aplicación IPTV en un entorno Web real.

Fig. 15. ANCHO DE BANDA UTILIZADO POR LA REPRODUCCIÓN DE UN CONTENIDO



Fuente: los autopres

IX. CONCLUSIONES

La arquitectura para el soporte del servicio de VoD apoyada en interactividad y sistemas de recomendaciones provee aportes en el área de la telemática como la convergencia de tecnologías, técnicas, protocolos, lenguajes y plataformas como *Java*, *HTML*, *Python*, *JSON*, *REST* y *OWL*, para ofrecer servicios de la web 2.0 en el contexto de VoD en IPTV. Además permite: el despliegue de servicios de apoyo, por ejemplo valoraciones, *chat* y recomendaciones con el fin de mejorar la experiencia del usuario en el uso del servicio de VoD; y la generación de conocimiento colaborativo en torno a un contexto específico, en este caso el cine.

Con el objetivo de generar recomendaciones más acertadas, posibilita el uso de servicios que permiten coleccionar información acerca de las preferencias de los usuarios. Por ejemplo, a través de la actividad en el *chat*, es posible inferir un porcentaje del nivel de interés de un usuario por un contenido sin que éste tenga que especificarlo a través del servicio de valoraciones. Debido a que

existe el riesgo de que se afecte la privacidad de los usuarios, es necesario el uso de un mecanismo por medio del cual se acepten los términos de dicha inferencia.

Esta arquitectura puede utilizarse en entornos comerciales de demanda de películas y series de televisión. Desde el punto de vista académico, puede dar soporte a Comunidades Académicas Virtuales, en las que los miembros pueden generar y compartir contenidos educativos e intercambiar y construir conocimiento sobre un tema específico.

El prototipo del servicio se validó a través de pruebas de rendimiento y ancho de banda en el Laboratorio de Televisión Digital de la Universidad del Cauca en un entorno controlado del canal. Como resultados, se estimó que en un escenario real el sistema podría responder a las peticiones de los usuarios de forma rápida en situaciones de estrés (elevado número de peticiones simultáneas). Esto, debido al uso de los principios de *REST* en la arquitectura, definidos para un consumo mínimo de recursos. De la prueba de ancho de banda, se encontró un comportamiento acorde al de un servicio sobre una red no gestionada: no se asegura QoS.

AGRADECIMIENTOS

Este proyecto se desarrolló con el apoyo de la Universidad del Cauca-Colombia, los proyectos STCAV (Servicios de *t-Learning* para el Soporte de una Comunidad Académica Virtual), GESTV (Plataforma de Gestión para un Sistema de *t-Learning*), Redauti y el grupo de investigación del Laboratorio de Televisión Digital de Vigo-España.

REFERENCIAS

- [1] J. Hjelm, *Why IPTV?, Interactivity, Technologies and Services*. UK: Wiley, 2008, p. 358.
- [2] F. Moumtadi, M. Escobar, R. Argota, y S. Landeros, "Reducción del retardo en el cambio de canal en servicios IPTV," *Ingeniería, Investigación y Tecnología*, vol. 9, No. 3, pp. 217-229, Ene. 2008.
- [3] R. Turrin and P. Cremonesi, "Recommender Systems for Interactive TV," *8th European Conference on Interactive TV and Video Tutorial*, Jun. 2010.
- [4] S. Vogl, P. Halbmayer, and C. Lichtenberger, "Media Space Navigator: Navigating Video Content on IPTV"

- Portals,” *Lecture Notes in Computer Science*, vol. 5066, pp. 248-252, Jul. 2008.
- [5] C. Zheng and Z. Ming, “An efficient video similarity search strategy for video-on-demand systems,” *Broadband Network & Multimedia Technology, IC-BNMT '09. 2nd IEEE International Conference*, pp. 174-178, Oct. 2009.
- [6] C. Tan, G. Chang, X. Wang, and C. Zhu, “An EPG Recommender Framework with Information Exchange between Users,” *Hybrid Intelligent Systems, HIS '09, ninth International Conference*, vol. 2, pp. 453-456, Aug. 2009.
- [7] B. Xiao, J. Yan, X. Guo, and L. Leung, “IPTV: User Behavior Analysis,” *Management and Service Science, 2009. MASS '09. International Conference*, pp. 1-4, Sep. 2009.
- [8] M. Ahmad, J. Qadir, N. Rehman, A. Baig, and H. Majeed, “Prediction-based channel zapping latency reduction techniques for IPTV systems – A survey,” *Emerging Technologies, 2009. ICET 2009. International Conference*, pp. 466-470, Oct. 2009.
- [9] M. Kim, S. Min Ko, J. Mun, G. Ji, and M. Jung, “A USA-BILITY STUDY ON PERSONALIZED EPG (PEPG) UI OF DIGITAL TV,” *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 4552, pp. 892-901, jul. 2007.
- [10] Y. Xue and Q. Gong, “Study on Internet Recommendation System of Collaborative Filtering Based on Scatter Diference,” *Computer, Mechatronics, Control and Electronic Engineering (CMCE), International Conference*, vol. 1, pp. 160-163, Aug. 2010.
- [11] K. Lee, K. Cho, and W. Ryu, “Social TV Service: A case study,” *Consumer Electronics (ICCE), IEEE International Conference*, pp. 287-288, Jan. 2011.
- [12] E. Kim and S. Ko, “Investigating User Adoption of T-commerce,” *Computers, Networks, Systems and Industrial Engineering (CNSI), First ACIS/JNU International conference*, pp. 95-99, May. 2011.
- [13] UK Government Digital Television. Disponible <http://www.digitaltelevision.gov.uk>. Consultado el 1 de Julio de 2012.
- [14] G. Held, *Understanding IPTV*. New York: Auerbach publications, 2007, p. 189.
- [15] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-Based Collaborative Filtering Recommendation,” *Proceedings of the 10th international conference on World Wide Web*, pp. 285-295, May 2005.
- [16] W. Campo, G. Chanchí, y J. Arciniegas, “Arquitectura para el Soporte de Comunidades Académicas Virtuales en Ambientes de TDi”, Universidad del Cauca, Colombia, 2010.
- [17] Proyecto ST-CAV, disponible en: <http://www.unicauca.edu.co/stcav/>. Consultado el 1 de Julio de 2012.
- [18] J. Lloret, M. García, F. Boronat, “IPTV: la televisión por Internet”, Editorial Vértice, Málaga, España, 2008, p. 230.
- [19] Y., Blanco, *Propuesta Metodológica para el Razonamiento Semántico en Sistemas de Recomendación Personalizada y Automática*. Vigo, España: Universidad de Vigo, 2007.
- [20] M. Björkman, L. Aroyo, M. Bellekens, T. Dekker, E. Loeff and R. Pulles, “Personalised Home Media Centre Using Semantically Enriched TV-Anytime Content,” *4th European Conference on Interactive Television (EuroITV-06)*, pp. 165-173, May 2006.
- [21] X. Li, L. Xie, and G. Liu, “Study of VOD Server Architecture with the Array of the Balanced Nodes in Storage and Networks,” *Frontier of Computer Science and Technology (FCST), 2010 Fifth International Conference*, pp. 214-218, Aug. 2010.
- [22] O. Friedrich, R. Seeliger, A. Al-Hezmi, C. Riede, and S. Arbanowski, “Prototyping Interactive and Personalized IPTV-Services on Top of Open IMS Infrastructures,” *8th European Conference on Interactive TV and Video*, pp. 204-208, Jun. 2010.
- [23] The Protégé Ontology Editor and knowledge Acquisition System. Disponible: <http://protege.stanford.edu>. Consultado el 1 de Julio de 2012.
- [24] Python Programming Language. Disponible: <http://www.python.org/>. Consultado el 1 de Julio de 2012.
- [25] Apache Jena. Disponible: <http://jena.apache.org/>. Consultado el 1 de Julio de 2012.
- [26] J. Nielsen, *Usability Engineering*. San Francisco, USA: Morgan Kaufman, 1994.